# Secure Cloud Storing and Sharing of Big Data Using SSGK Technique

Konduru Anusha[1], Mayana Atheequllah Khan[2]
[1]*M. Tech Student, Dept of Computer Science and Engineering, Sri Sai Institute of Technology and Science, Rayachoti, Andhra Pradesh, India*
[2]*Associate Professor, Dept of Computer Science and Engineering, Sri Sai Institute of Technology and Science, Rayachoti, Andhra Pradesh, India*

***Abstract* - A cloud-based big data sharing system utilizes a storage facility from a cloud service provider to share data with legitimate users. In contrast to traditional solutions, cloud provider stores the shared data in the large data centers outside the trust domain of the data owner, which may trigger the problem of data confidentiality. This paper proposes a secret sharing group key management protocol (SSGK) to protect the communication process and shared data from unauthorized access. Different from the prior works, a group key is used to encrypt the shared data and a secret sharing scheme is used to distribute the group key in SSGK. The extensive security and performance analyses indicate that our protocol highly minimizes the security and privacy risks of sharing data in cloud storage and saves storage space.**

***Index Terms* - Big data, security, privacy, cloud storage, data sharing.**

## I.INTRODUCTION

Emerging technologies about big data such as Cloud Computing, Business Intelligence, Data Mining, Industrial Information Integration Engineering (IIIE) and Internet-of Things have opened a new era for future Enterprise Systems (ES). Cloud computing is a new computing model, in which all resource on Internet forms a cloud resource pool and can be allocated to different applications and services dynamically. Compared with traditional distribute system, a considerable amount of investment saved, and it brings exceptional elasticity, scalability, and efficiency for task execution. By utilizing Cloud Computing services, the numerous enterprise investments in building and maintaining a supercomputing or grid computing environment for smart applications can be effectively reduced. Despite these advantages, security requirements dramatically rise when storing personal identifiable on cloud environment. This raises regulatory compliance issues since migrate the sensitive data from federate domain to distribute domain. To take the benefit enabled by big data technologies, security and privacy issues must be addressed firstly. Building security mechanism for cloud storage is not an easy task. Because shared data on the cloud is outside the control domain of legitimate participants, making the shared data usable upon the demand of the legitimate users should be solved. Additionally, increasing number of parties, devices and applications involved in the cloud leads to the explosive growth of numbers of access points, which makes it more difficult to take proper access control. Lastly, shared data on the cloud are vulnerable to lost or incorrectly modified by the cloud provider or network attackers. Protecting shared data from unauthorized deletion, modification and fabrication is a difficult task.

Online backup systems are classically built a client software application that run on a program determined by the purchase stage of service. Cloud backups contain the software and hardware component to keep an organization's data, include applications Exchange and SQL Server. Online backup is used by small and medium sized businesses (SMBs) and larger enterprises to back up the data. For larger organization, cloud data backup as a complementary form of backup.

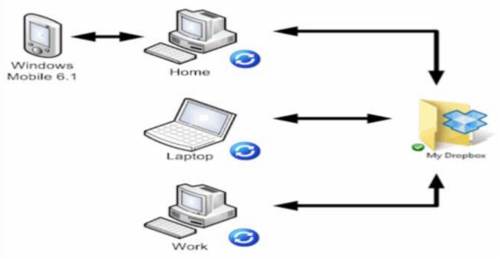

Figure 1: Cloud Backup Services

normally compensate for their data storage on cloud as per-usage or monthly rate. The cloud Storage providers provide a platform as a service, is one of the infrastructure services on cloud storage to shorten storage management for enterprises and personality users. Implementing cloud data backup is able to help boost an organization data protection without raising the workload on information technology.

## II. LITERATURER SURVEY

Distributed File Systems:

SSGK PROTOCOL is an intrusion-tolerant file scheme that maintains data privacy, integrity, and availability regardless of the existence of compromised components. Our design adopts some ideas from presented file systems, such as the severance of data and metadata from NASD, volume lease from AFS cloud services instead of communicating directly for coordination.

Data-centric coordination:

A key feature of SSGK PROTOCOL is the use of Byzantine-resilient datacentric algorithms for implementing storage space and coordination. There are some works that suggest the use of this kind of algorithms for apply dependable systems [15], [26], [39]. Byzantine disk Paxos [26] is a consensus protocol built on top of untrusted shared disks. More freshly, an enhanced version of this protocol specifically designed to use file synchronization services (e.g., DropBox, Google Drive) instead of disks was published [21]. To the best of our knowledge, there are only two fault-tolerant data-centric lease algorithms in the literature [15] compare with SSGK PROTOCOL 's BFT composite lease.

Multi-cloud storage:

In the last years, many works have been proposing the use of multiple cloud providers to improve the integrity and availability of stored data [13], [14], [15], [16], [20], [22], [23], [24], [70]. A problem in some of them is the fact they only provide object storage (i.e., read/write registers), which hardens their integration with existing applications. Examples of these systems are RACS [13] for writeonce/archival storage, and DepSky [15], its evolution [16], and ICStore [14] for updatable registers. The main limitation of these systems is that they require servers deployed in the cloud providers, which implies additional costs and management complexity.

FRUGAL CLOUD FILE SYSTEM:

Different embodiments grant a techniques and tools of providing a frugal cloud file system [5] that proficiently uses the blocks of different types of storage devices with special properties for various purposes. The various types of storage strategy reduce the storage and bandwidth transparency. Favorably, the storage and bandwidth reduction in the clouds achieved by the frugal cloud file system, reduce the cost-effective of managing the file system at the same time, maintain high performance. Frugal file system is a structure that optimizes on the whole storage cost between various Cloud storage services and different type of price. The Frugal file system's storage services like a twin storage system, one is low latency (e.g., Amazon ElastiCache) and the other one is high latency (e.g., Amazon S3). In low latency the data transfer cost is low and cost of storage per byte is high (i.e., cache) but the high latency, cost of storage per byte is low and data transfer cost is high (i.e., disk) Distributed file system (DFS) is the file systems for cloud. DFS allows the clients to access data. Data files are separated by parts as chunks that stored on various remote systems which offer the parallel execution. Data are stored in files in the format of hierarchical tree structure. Directories are denoted by nodes. DFS facilitate any type of enterprises (such as large, medium, small) that allows storing the data and accessing the data on remotely. DFS allow two type of file system as GFS and HDFS. Both file systems are handled the batch processing. Hadoop distributed file system (HDFS) designed for access the terabyte's data or peta bytes data. HDFS is master slave architecture. It consists of Name node and Data node mechanisms. Name node manages the storage of Metadata and Data Node manages the node storage. HDFS file systems the files are divided into blocks. Each block contains various data nodes, and every node is replicated for availability. This is the block level replication. Name node manages the operation of name space and map the block to data node. HDFS is characterized by method of in spite of their increasing Technology, current cloud backed storage systems still have various restrictions related to reliability, durability assurances and inefficient file sharing. SCFS, a cloud-backed file system overcome these challenges and provides strong consistency and POSIX semantics for cloud backed services. It uses a plug gable back plane allows the various cloud storage or a cloud-of-clouds. The main goal of SCFS is assurance of security like integrity, confidentiality, availability and also supports consistency-on-close semantics [28], SCFS is not proposed to be a big-data file system, because file data is downloaded from and uploaded to one or more clouds. SCFS contain the backend cloud storage, co-ordination service and SCFS agent. File data are maintained by backend cloud storage. Metadata management and synchronization supported by co-ordination service. SCFS functionality and client file system mounted by SCFS agent. SCFS provide the strong cloud consistency based on two approaches as maintaining the Meta with limited capacity data and save the data itself. SCFS support two prototype coordination services: zookeeper [29] and Depspace [13]. The two services are integrated with SCFS wrappers. The co –ordination services simulated the fault tolerance. Zookeepers need 2f + 1 replica for tolerate f crashes by using Paxos-like protocol. Depspace need 3f + 1 or 2f + 1 replica for tolerate f arbitrary faults by using BFT-SMaRt replication engine. SCFS cloud storage services are Amazon S3, Windows Azure Blob, Google Cloud Storage, Racks pace cloud files and all services using cloud of cloud back end. Cloud back end use the extended version of DepSky, support a new operation as read all versions of hashes data are stored in Depsky's Metadata internal objects and stored in cloud. Based on the consistency and sharing requirements of stored data the SCFS operation divided into 3 modes. 1) Blocking mode 2) Non –blocking mode 3) Non- sharing mode. The main uses of SCFS are backup, disaster recovery, and file sharing control and without require dependence on any single cloud provider.

DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds DEPSKY, cloud backup improves the availability, integrity and confidentiality of information store in the cloud with help of encoding, replication and encryption of the data on varied clouds that make a cloud-of-clouds. DEPSKY is a reliable and protected storage system that gives the profit of cloud computing by using an arrangement of diverse commercial clouds to cloud-of-clouds. DEPSKY also give the virtual storage, it is accessed by users while invoking the operations. DEPSKY also provide the four limitations such as Loss and corruption of data, Loss of privacy, Vendor lock-in, Loss of availability. DEPSKY System use data and system models. It contains two main algorithms as DEPSKY – A and DEPSKY – CA and also contains the set of auxiliary protocols. Two algorithms are implemented by software library in the clients. Data models contain the three abstraction levels. First level, the conceptual data unit has unique name, version number – support the object updates, data verification – a cryptographic data hash. Second level, Conceptual unit is implemented by generic data unit, has two types of files: signed Metadata file and storage file. Third level, data unit are implemented. Data unit support the operation of storage objects like creation of Metadata file, destruction of data unit, write operation and read operation. System Model uses the asynchronous distributed system; it is composed by writers, readers, and cloud storage providers. Quorum protocols can provide as the backbone of storage systems. Quorum protocols contain the individual storage nodes instead of servers. Many protocols involve several steps to access the shared memory, it makes unrealistic for geographically isolated distributed systems such as DEPSKY. The DEPSKY protocols need two communication roundtrips to read or write the metadata and data files. Byzantine fault tolerant (BFT) storage is implemented by several protocols. But it requires server for execute code and functions; it is not available on cloud storage. This the key difference between DEPSKY protocols and BFT protocols. DEPSKY – A is the protocol of DEPSKY, it improves the availability and integrity of storage cloud by replication using quorum techniques. DEPSKY -A include read and write algorithm the DEPSKY-A protocol has two major restrictions. First, one is data unit size and costs, the data stored in single cloud. Second one is data storage is clear text, so it does not provide confidentiality guarantee. DEPSKY – CA protocol overcome these problems and also has the additional cryptographic function and coding functions. DepSky-CA write algorithm's encryption technique generates the key sharing.

## III. PROPOSED SYSTEM

In this proposed system common temp key is shared to reduce the information leakage from cloud storage in big data. To minimize security and privacy risks some limits were provided which are time limit, size limit, and credit point limit. Information was encrypted to provide more security (AES, DES algorithm). The temp key can be used by person who requests to retrieve information for once. If other than the request person tries to use temp key, then that key is removed, and alert notification will be sent to data owner. Temp key provider sends the key to request person by mail using SMTP protocol. (Gmail -high secure) The main advantage of proposed system is to separate storage space into module and each module is secured with temp password. This makes more efficient constructions. This key can be used only once. We propose a new secret sharing scheme that is computationally secure and can reduce the number of shares Temp key helps information retrieval more secured with low cost. Only request person can use temp key. Encryption standards make information difficult to theft. Limitations of temp key provide high security.

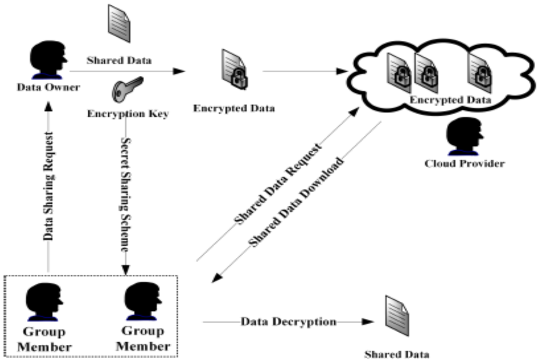

Fig 1. System Architecture

Cloud storage provides not just low cost, but high scalability and availability. It may be a natural solution to some of problems in storing and analyzing the increasing patients' medical records. For healthcare providers, only based on the aggregation of all patients' medical records, could proper diagnosis be made. Reference proposed a cloud-based platform for healthcare. Cloud storage provides a common place for storing medical records which overcome the delay of transferring medical records between different healthcare providers and make diagnostic process more efficient. The e-healthcare cloud provides many advantages in collaboration and data sharing among healthcare providers. Nevertheless, in consider of the highly privacy of medical data it comes with significant risks of medical records. Firstly, medical records are shared on the public channel where many attackers on the channel to eavesdrop the medical records. Additionally, due to the increasing number of parties, devices and applications involved in cloud, unauthorized parties or cloud providers may have the ability to access shared medical records. Last but not the least, some authorized parties may work together to get some unauthorized medical records illegally. E-healthcare services require a security mechanism to protect the privacy of medical records.

## IV METHODOLOGY

In this section, we define the application scenario of our protocol and security requirements.

A. CLOUD STORAGE FOR BIG DATA The architecture of cloud based big data is illustrated in Figure.1. It consists of three parts: source data, cloud center and services. Between source data and cloud center layer, unstructured or semi-structured source data is structured. They include processing methods such as data collection [3], data mining [3] and data aggregation [3]. The processed source data is stored on cloud in relational or NoSQL databases [3]. Lastly, service layer answers information requests submitted by consumers by integrating information stored in cloud. Beyond allowing customers to put all data into cloud, cloud storage provides all kinds of data services for customers. Because scale horizontally runs on cheap commodity hard in a distributed configuration and there is no need for customers to purchase and maintain their own IT facilities, cloud based big data stores brings in inherent availability, scalability, and cost effectiveness.

B. AN EXAMPLE OF HEALTHCARE INFORMATION SYSTEM Cloud storage provides not just low cost, but high scalability and availability. It may be a natural solution to some of problems in storing and analyzing the increasing patients' medical records [7]. For healthcare providers, only based on the aggregation of all patients' medical records, could proper diagnosis be made. Reference [8] proposed a cloud-based platform for healthcare. Cloud storage provides a common place for storing medical records which overcome the delay of transferring medical records between different healthcare providers and make diagnostic process more efficient. The e-healthcare cloud provides many advantages in collaboration and data sharing among healthcare providers. Nevertheless, in consider of the highly privacy of medical data it comes with significant risks of medical records. Firstly, medical records are shared on the public channel where many attackers on the channel to eavesdrop the medical records. Additionally, due to the increasing number of parties, devices and applications involved in cloud, unauthorized parties or cloud providers may have the ability to access shared medical records. Last but not the least, some authorized parties may work together to get some unauthorized medical records illegally. E-healthcare services require a security mechanism to protect the privacy of medical records. In this section we describe more about the proposed protocol model and algorithm of SSGK. A. PROTOCOL MODEL

1) DATA SHARING MODEL Consider a cloud storage data sharing system with multiple entities and the data sharing model is shown as Figure.2. The protocol model consists of three types of entities: cloud provider, data owner and group members. The cloud provider provides a public platform for data owners to store and share their encrypted data. The cloud provider does not conduct data access control for owners. The encrypted data can be download freely by any users. FIGURE 1. Data protocol model of the proposed SSGK. Data owner: defines the access policy and encrypts its data with a symmetric encryption algorithm using a group key. The group members who satisfied the access policy constitute a sharing group. Then secret sharing scheme is used by the owner to distribute the encryption key to the sharing group. Group members: every group member including the data owner is assigned with a unique and a pair of keys. The group members can freely get any interested encrypted data from the public cloud. However, the user can decrypt the data if and only if it gets the data decryption key from the data owner.

2) SECURITY MODEL In SSGK, we have the following assumptions: The data owner is totally trusted and will never be corrupted by any adversaries. Cloud provider is semi-trusted, it correctly executes the task assigned to them for protests, but they would try to and out as much secret information as possible based on the data owners uploaded data. We now describe the security model of SSGK by listing possible attacks. The group key is distributed by running the secret sharing scheme. Parts of the group members can gather their sub-secret shares to reconstruct the group key. Moreover, the communication channel of our protocol is defined as: Every pair of participants have a point-to-point channel to send messages. Additionally, all the participants access to a broadcast channel: when a participant puts a message m on this channel, all the other participants receive m. The group key is distributed on the public channel and the key may be tempered by adversaries. Verify: A verification algorithm that, on input a sub-share and v, output whether the sub-share is tempered during distribution; Secret Reconstructed: For any t sub-shares, the security parameter K can be reconstructed. Definition 2 (Equity and Availability): Verified secret sharing scheme guaranteeing equity and availability with two conditions: Any participant set in the share group, where the size of the set is less than the total quantity, the participants in the set cannot get any information about K; Only with cooperation of all the legitimate participants, K could be reconstructed. Definition 3 (Confidentiality): Verified secret sharing scheme guarantees confidentiality if any users outside the sharing group cannot get any information of K even with the knowledge of enough interactive messages. Definition 4 (Integrity): Once the interactive messages are tempered during VSS, any information about K could be gotten by participants. We said that verified secret sharing scheme guarantees integrity. The notations in Table 1 are used throughout the remainder of this paper. In this paper, we propose a novel group key management protocol for the data sharing in the cloud storage. In SSGK, we use RSA and verified secret sharing to make the data owner achieve ne-grained control over the out-sourced data without relying on any third party. In addition, we give detailed analysis of possible attacks and corresponding defenses, which demonstrates that GKMP is secure under weaker assumptions. Moreover, we demonstrate that our protocol exhibits less storage and computing complexity. Security mechanism in our scheme guarantees the privacy of grids data in cloud storage. Encryption secures the trans-mission on the public channel; verified security scheme make the grids data only accessed by authorized parties. The better performance in terms of storage and computation make our scheme more practical. The problem of forward and backward security in group key

management may require some additions to our protocol. An efficient dynamic mechanism of group members remains as future work.

## V. CONCLUSION

Common temp key is shared to reduce the information leakage from cloud storage in big data. To minimize security and privacy risks some limits were provided which are time limit, size limit, and credit point limit. Information was encrypted to provide more security (AES, DES algorithm). The temp key can be used by person who requests to retrieve information for once. If other than the request person tries to use temp key, then that key is removed, and alert notification will be sent to data owner. Temp key provider sends the key to request person by mail using SMTP protocol. (Gmail -high secure) The main advantage of proposed system is to separate storage space into module and each module is secured with temp password. This makes more efficient constructions. This key can be used only once. We propose a new secret sharing scheme that is computationally secure and can reduce the number of shares Temp key helps information retrieval more secured with low cost. Only request person can use temp key. Encryption standards make information difficult to theft. Limitations of temp key provides high security.

## REFERENCE

[1] Future of cloud computing - 2nd annual survey results. http://goo.gl/fyrZFD, 2012.

[2] S3FS-FUSE-based file system backed by Amazon S3.

[3] http://code.google.com/p/s3fs/.

[4] S3QL - a full-featured file system for online data storage.

[5] http://code.google.com/p/s3ql/.

[6] http://searchcloudstorage.techtarget.com/definiti on

[7] Krishna P.N. Puttaswamy, Thyaga Nandagopal and Murli kodialam "Frugal storage for cloud file system," in proceeding EuroSys'12 of the 7th ACM European conference on computer Systems, 2015 pages 71-84.

[8] https://en.wikipedia.org/wiki/Distributed

[9] Michael Vrable, Stefan Savage, and Geoffrey M. Voelker "Blue Sky: A Cloud-Backed File System for the Enterprise, "in Proceedings of the 10th USENIX conference on File and Storage Technologies, Feb 2012.

[10] M. Rosenblum and J. K. Ousterhout." The Design and Implementation of a Log-Structured File System," ACM Transactions on Computer Systems (TOCS), 1992.

[11] Alysson Bessani, Ricardo Mendes, Tiago Oliveira, Nuno Neves, Miguel Correia, Marcelo Pasin, and Paulo Verissimo "SCFS: A Shared Cloud-backed File System, "in Proceedings of the USENIX ATC on USENIX Annual Technical Conference 19&20-June -2014.

[12] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish "Depot: Cloud Storage with Minimal Trust," In Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI), Oct. 2010.

[13] J. Howard "Scale and performance in a distributed file system" ACM Trans. Computer Systems, 1988.

[14] P. Hunt, M. Konar, F. Junqueira, and B. Reed. "Zookeeper: Waitfree coordination for internet-scale services," In USENIX ATC, 2010.

[15] A. Bessani, E. P. Alchieri, M. Correia, and J. S. Fraga "DepSpace: A Byzantine fault-tolerant coordination service," in EuroSys, 2008.

[16] StorSimple. StorSimple. http://www.storsimple. com/.

[17] TwinStrata. TwinStrata. http://www.twinstrata. com/.

[18] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando Andr´e, and Paulo Sousa "DEPSKY: Dependable and Secure Storage in a Cloud-of-clouds," EuroSys 11-April- 2011.

[19] Ricardo Mendes, Tiago Oliveira, Vinicius Cogo, Alysson Bessani "The SSGK PROTOCOL file system,"

[20] Idilio Drago, Marco Mellia, Maurizio M. Munafò, Anna Sperotto and Aiko Pras "Inside Drop box: Understanding Personal Cloud Storage Services," in Proceeding of IMC -12 of ACM conference on internet measurement conference,2012 PP.481-494.

[21] http://www.techrepublic.com/blog/five-apps/ keep-your-data-safewith-one-of-these-five-cloud -backup-tools/

[22] http://www.cloudwards.net/spideroak-or-wuala-which-is-moresecure/

[23] Kailas Pophale, Priyanka Patil, Rahul Shelake, Swapnil Sapkal "Seed Block Algorithm: Remote Smart Data- Backup Technique for Cloud Computing," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 3, March 2015.

[24] Lili Sun, Jianwei An, Yang, and Ming Zeng "Recovery Strategies for Service Composition in Dynamic Network," International Conference on Cloud and Service Computing,2011

[25] Giuseppe Pirr´o, Paolo Trunfio, Domenico Talia, Paolo Missier and Carole Goble "ERGOT: A Semantic-based System for Service Discovery in Distributed Infrastructures," 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.

[26] Xi Zhou, Junshuai Shi, YingxiaoXu, Yinsheng Li and Weiwei Sun "A backup restoration algorithm of service composition in MANETs," Communication Technology ICCT 11th IEEE International Conference, 2008, pp. 588-591.

[27] Ms.KrutiSharma, and Prof K.R. Singh "Online data Backup and Disaster Recovery techniques in cloud computing: A review", JEIT, Vol.2, Issue 5, 2012.

[28] Eleni Palkopoulouy, Dominic A. Schupke, Thomas Bauscherty "Recovery Time Analysis for the Shared Backup Router Resources (SBRR) Architecture", IEEE ICC, 2011.

[29] http://searchcloudstorage.techtarget.com/definition/cloud-storage

[30] G. R. Blakley, ''Safeguarding cryptographic keys,'' in Proc. AFIPS 79th Nat. Comput. Conf., Jun. 1979, pp. 313 317.

[31] A. Shamir, ''How to share a secret,'' Commun. ACM, vol. 22, no. 11, pp. 612 613, Nov. 1979.

[32] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, and S. sharing, ''Variable secret sharing and achieving simultaneity in the presence of faults,'' in Proc. 26th IEEE Symp. Found. Comput. Sci., Oct. 1985, pp. 383 395.

[33] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, ''Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey,'' IEEE Commun. Surveys Tuts., vol. 18, no.4, pp. 2546 2590, 4th Quart., 2016.

[34] A. L. Buczak and E. Guven, ''A survey of data mining and machine learning methods for cyber security intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153 1176, 2nd Quart., 2016.

[35] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, ''Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks,'' IEEE Trans. Dependable Secure Comput., vol. 12, no. 1, pp. 98 110, Jan./Feb. 2015.

[36] J. R. Lourenço, V. Abramova, B. Cabral, J. Bernardino, P. Carreiro, and M. Vieira, ''No SQL in practice: A write-heavy enterprise application,'' in Proc. IEEE Int. Congr. Big Data, Jun./Jul. 2015, pp. 584 591.

[37] V. Casola, A. Castiglione, K. K. Choo, and C. Esposito, ''Healthcare-related data in the cloud: Challenges and opportunities,'' IEEE Cloud Comput., vol. 3, no. 6, pp. 10 14, Apr. 2016.